

©2020 IEEE. Personal use of this material is permitted. Permission from IEEE must be obtained for all other uses, in any current or future media, including reprinting/republishing this material for advertising or promotional purposes, creating new collective works, for resale or redistribution to servers or lists, or reuse of any copyrighted component of this work in other works.

# Incorporating Spatial Constraints into a Bayesian Tracking Framework for Improved Localisation in Agricultural Environments

Muhammad W. Khan\*, Gautham P. Das\*, Marc Hanheide\* and Grzegorz Cielniak\*

**Abstract**—Global navigation satellite system (GNSS) has been considered as a panacea for positioning and tracking since the last decade. However, it suffers from severe limitations in terms of accuracy, particularly in highly cluttered and indoor environments. Though real-time kinematics (RTK) supported GNSS promises extremely accurate localisation, employing such services are expensive, fail in occluded environments and are unavailable in areas where cellular base stations are not accessible. It is, therefore, necessary that the GNSS data is to be filtered if high accuracy is required. Thus, this article presents a GNSS-based particle filter that exploits the spatial constraints imposed by the environment. In the proposed setup, the state prediction of the sample set follows a restricted motion according to the topological map of the environment. This results in the transition of the samples getting confined between specific discrete points, called the topological nodes, defined by a topological map. This is followed by a refinement stage where the full set of predicted samples goes through weighting and resampling, where the weight is proportional to the predicted particle's proximity with the GNSS measurement. Thus, a discrete space continuous-time Bayesian filter is proposed, called the Topological Particle Filter (TPF).

The proposed TPF is put to test by localising and tracking fruit pickers inside polytunnels. Fruit pickers inside polytunnels can only follow specific paths according to the topology of the tunnel. These paths are defined in the topological map of the polytunnels and are fed to TPF to track fruit pickers. Extensive datasets are collected to demonstrate the improved discrete tracking of strawberry pickers inside polytunnels thanks to the exploitation of the environmental constraints.

## I. INTRODUCTION

Safe robot behaviour is one of the most critical requirement of service robots in agri-robotics [1]. A robot or a fleet of robots needs to be aware of, not only the static environment like buildings and polytunnels but also dynamic subjects like humans and other robots. This is essential for safe navigation, collision avoidance, versatile manoeuvring and the availability of a robot for service on demand. The static environment can be mapped before operational robots are deployed via simultaneous localisation and mapping (SLAM) [2], [3]. Localising and tracking humans, however, presents new challenges for which the robots are equipped with different sensors. 2D and 3D lidars are the most widely used laser-based sensors used for localisation and tracking of humans [4], and are well known for their wide coverage angles that provide a 360-degree view of the surrounding. Radio-Frequency Identification tags (RFIDs) are also popular in short-range location estimation and can also provide the identity of the localised person. These sensors, however,

\*The authors are with the Lincoln Centre for Autonomous Systems at the School of Computer Science of University of Lincoln, United Kingdom.



Fig. 1: A setup for an autonomous fleet of robots to support fruit pickers. Call-A-Robot (CAR) user interface: (a) ready to call a robot, (c) robot on the way to picker, (d) robot arrived at picker's location and ready to be loaded, (f) robot sent to in-field storage with picked yield. A human picker picking strawberries (b) and loading full trays on a robot (e).

require the robot to be in the vicinity of humans, for the sensor to observe them and hence cannot provide continuous service. For continuous tracking of humans, it is therefore imperative that a Global Navigation Satellite System (GNSS) is used. In this paper, a GNSS-based spatially constrained particle filter, referred to as Topological Particle Filter (TPF) is proposed. The state of TPF follows an iterative update, constrained by the topology of the environment. The topology of the environment is captured by the topological map, which is a discrete representation of the environment and acts as an input to the TPF. Thus, this setup restricts the transition of particles only between discrete points, called the topological nodes, defined in the topological map. As a result, a particle can only transit from one topological node to another, if such transition is allowed by the topological map.

The proposed TPF is tested by tracking fruit pickers inside polytunnels. In our application, the fruit picker can request a robot at will. Thus, for the robot to reach the fruit picker, it needs to know the exact location from where the request was

made. An error in location with variance greater than  $0.75\text{m}^1$  in the picker's estimated location can send the robot to the wrong lane of a polytunnel, making it impossible for the fruit picker to access the robot physically. The TPF makes sure that the picker's estimated location is accurate enough for the robot to navigate to the topological node closest to the picker.

The main contributions of this work are as follows: 1) a spatially constrained particle filter is designed that enables the robots to track the workers throughout the operation time exploiting the constraints imposed by the environment, 2) a mobile app is designed that the workers can use to call, cancel and check the status of the incoming robot, 3) the full system is validated in a real fruit production scenario.

The proposed system was developed as part of the RAS-Berry (Robotics and Autonomous Systems for Berry production) programme [6] which aims to develop an autonomous fleet of robots to provide support to fruit pickers and other workers in horticulture industry. The number of applications range from fruit transportation, UV treatment of fungi to autonomous picking of ripe fruit. In this paper, we consider the in-field transportation with the main objective to provide fruit pickers with an autonomous robotic platform for the transportation of picked yield. Fruit pickers in polytunnels are required to send picked yield to a food handling unit (FHU) as quickly as possible. At FHU the yield is cleaned, packed and refrigerated before it is sent off to the markets. These handling units are often situated at a distance from the polytunnels where the pickers are picking. Furthermore, the fruit pickers need to finish picking in all polytunnels before the picked fruits are transported to FHU. Thus, transporting the yield from the polytunnels to the FHU takes a substantial amount of time, during which the yield is kept on trolleys out in the open. This significantly reduces the shelf life of the yield in the market. Thus, quick transportation of the yield to the FHU is of the essence. The core component of the system is a robust picker localisation system which is described and evaluated in the following sections.

## II. RELATED WORK

Detection, identification, localisation and tracking of objects has been an important subject studied in the robotics community [7], [8]. There are systems which have achieved sub-meter level accuracy but at the expense of high-cost sensors like lidars, high-definition cameras, RTK-supported GNSS or different combinations of the sensors. Both active and passive approaches to localisation and tracking are extensively studied in the literature.

One of the popular passive sensing approaches towards localisation relies on colour video streams. A colour camera can provide several local and temporal features, however, it fails to provide 3D location in real-world [9], [10], [11] and thus has a limited range of applications. Thus, depth cameras have received considerable attention since 2010 with the release of Microsoft Kinect. In [14], the authors proposed

an algorithm for the Kinect-equipped small-footprint robot which employs a legs classifier to detect and track the legs of multiple people in a highly cluttered indoor environment.

Passive sensors like lasers, cameras and RFIDs fail to provide the identity of localised objects, although some studies like [12], [13] have developed algorithms which can identify and track human subjects simultaneously. This type of identification is system generated and can only associate a track with a system-generated identifier. In [15], the authors used a 3D camera (Kinect V2) and a passive RFID (Impinj R420) to simultaneously track and identify humans. In their work, the R420's antenna behaves as a reference to match the RFID with the skeletons detected by the depth camera. Though a centimetre-level accuracy is achieved, the inability of Kinect V2 to detect more than six skeletons at once, confined the system's capacity to six tracks only in a relatively short range.

In [16], an online learning framework taking advantage of multi-target tracking using 3D-lidar is presented. The framework detects, identifies and tracks human subjects in real-time. However, as the case with any homogeneous laser-based system, the tracker is liable for degraded accuracy due to false positives and negatives. In [17], a robot self-localisation scheme is presented which fuses several heterogeneous sensors together and leverages their strengths by adding constraints on the shortcomings of each sensor. Using different sensor setups the authors were able to achieve from 37% to 76% accuracy improvements. In [18], multi-sensor human tracking with a mobile robot using different variants of the Kalman and particle filters is presented. The authors demonstrate that in terms of accuracy, the particle filter outperforms the two variants of Kalman filter but at the expense of computational load. Another comparison of Bayesian trackers, exploiting range estimation via received signal strength of a radio signal is studied in [19]. In this work, the accuracy of the particle filter is compared with and outperforms the extended and the conventional Kalman filters.

Most of these techniques require the tracked entity to be in the vicinity of the sensor providing the measurements. In applications where continuous human identification and tracking is required over a very large coverage area, these sensors fail to provide a reliable solution. In such scenarios, the use of GNSS-based service is still the most sensible option.

## III. SYSTEM OVERVIEW

Our project for robotic in-field logistics aims to speed up the delivery of picked yield from the polytunnels to the FHU by employing a fleet of yield carrying robots. Such a solution also reduces the workload on pickers, enabling them to pick more fruit in less amount of time, thus cutting the labour cost. In this section, some of the tools designed to achieve this goal are reported.

<sup>1</sup>The standard width of a polytunnel lane is 1.5m.

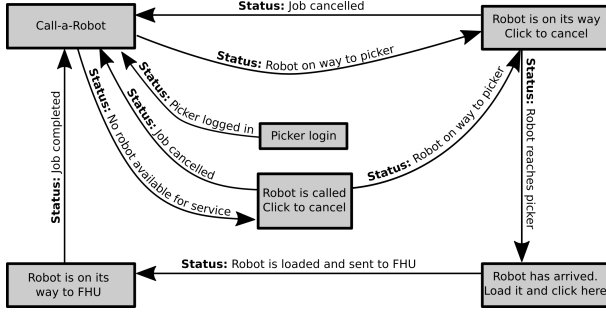


Fig. 2: An example state transition diagram for the CAR system.

#### A. CAR System

The CAR or Call-A-Robot system is an online mobile-based application that a fruit picker can use to request, cancel or to get a status of a particular transportation robot. Once logged in, the CAR application automatically sends the identity, timestamp and GNSS coordinates of the fruit picker to the robot via a coordination system. These coordinates are obtained from the mobile phone's internal GNSS or an external, more accurate GNSS module. The accuracy and update frequency of the coordinates depend on the GNSS module used. The CAR state transition diagram for one fruit transportation task is given in Fig. 2. Once the picker loads the trays and sends the robot to FHU, another call can be made.

#### B. Topological Map

A topological map (Fig. 4) is a discrete representation of the environment that can be viewed as a tuple  $T \rightarrow \langle N, E, A, Nav \rangle$ , where  $N$  is a set of discrete physical locations,  $n_j$ , called topological nodes i.e.,  $N = \{n_1, n_2, \dots\}$ .  $E \subseteq N \times N$ , represents the set of possible edges connecting the topological nodes, where the element at  $j^{th}$  row and  $k^{th}$  column of  $E$  is defined as

$$e_{jk} := \begin{cases} 1 & \text{if } n_j \text{ connects to } n_k \\ 0 & \text{elsewhere} \end{cases}, \quad (1)$$

where  $A$  and  $Nav$  are the sets of possible navigation actions performed by the robot and the mapping of each edge to a navigation action, respectively.  $N$  and  $E$  act as inputs to the TPF for the constraint prediction of the particles.  $A$  and  $Nav$  are utilised in topological navigation by the robots as described in detail in [21].

### IV. TOPOLOGICAL PARTICLE FILTER

Unlike the conventional particle filter, TPF tracks the picker's closest topological node, rather than the picker itself. The prediction step involves the transition of particles from node  $n_j$  to node  $n_k$ , only if  $n_j$  and  $n_k$  are connected through an edge  $e_{jk} \in E$ . This is to ensure that the movement of particles follow the topological map of the polytunnel. Thus the samples of TPF cannot jump from one lane to another, although for practical purposes this constraint is relaxed as

described in Section V. Every particle,  $p_i$  is associated with a probability of transition,  $r_i$ , which is a function of the particle's duration at a particular topological node and which resets to zero once a transition to another node occurs. The transition probability of  $i^{th}$  particle at  $j^{th}$  topological node is given by

$$r_i(\tau_i^j) = 1 - \exp(-\lambda \tau_i^j), \quad (2)$$

where  $\tau_i^j$  is the duration of  $i^{th}$  particle at topological node  $j$ . Thus, in the prediction step, the probability of a particle's transition from one topological node to another follows an exponential distribution. As a result, those particles with large  $\tau_i^{(\cdot)}$  values are more likely to transition to a connected topological node. The parameter  $\lambda$  is the tuning parameter and depends upon the speed of fruit picker. Particles associated with a fast picker need to transit more quickly between nodes as compared to particles associated with a slow-moving picker. The tuning parameter ensures that the particles follow a similar motion as the movement of fruit picker. The value for  $\lambda$  can be dynamic and calculated from the velocity obtained from the GNSS readings. This paper, however, assumes a constant value for all fruit pickers.

The correction step of TPF is two-fold and involves, obtaining weights for all predicted samples based upon the measurement from the GNSS module and its proximity to the predicted samples, resulting in a weighted predicted sample set. This results in every topological node acquiring a topological mass, which is the product of the number of predicted particles in that node and their associated weights. The closest node is then selected as the node with maximum topological mass. This is followed by a resampling step, where new particles are drawn with repetition from the predicted sample set, where the probability of selecting a particle is proportional to its weight.

The samples of the particle filter will not navigate between two lanes as no edge in  $E$  connects the topological nodes of any two lanes (see Fig. 4). Except at the entrance of the polytunnel where the topological nodes from different lanes are connected (white edges), demonstrating the possible movement of pickers from one lane to another. In this way, the samples of TPF always remains in the topological nodes belonging to the lane where the picker is picking. For this concept to work, the particle filter must be initialised in the correct lane. A wrong initialisation will result in all of the particles reside in the wrong lane and will never recover back to the lane where the picker is picking. To solve this problem, only a small portion of the particle set is allowed to move freely between topological nodes irrespective of an edge connection between the nodes. This ensures that the particle filter can gradually recover back to the correct lane if initialised in the wrong lane. This, however, comes at a cost. A consistent bias in the GNSS reading will allow the TPF to jump back to the wrong lane despite a correct initialisation. This concept is demonstrated with example in Section V.

## The TPF Framework

The step-by-step implementation of the TPF is explained in this section, for which the following notations are defined:  $g_t$  is the GNSS reading at time-step  $t$ ,  $p_t^i(j)$  and  $\hat{p}_t^i(k)$  is the representation of  $i^{th}$  particle located at  $j^{th}$  topological node and its predicted state at  $k^{th}$  topological node at time  $t$ , respectively.  $\mathbf{P}_t$ ,  $\mathbf{R}_t$ ,  $\hat{\mathbf{P}}_t$  and  $\mathbf{W}_t$  are the sets of all particles, their probabilities of transition, their predicted states and their associated weights, respectively. Total number of particles are denoted by  $N_s$ .

**0) Initialisation:** Create  $\mathbf{P}_{t=0}$  by generating  $p_{t=0}^i(\cdot) \in N \forall i$  s.t.  $p_{t=0}^i(\cdot) \sim \mathcal{N}(g_{t=0}, \sigma^2)$ .

*Explanation:* Initialisation is performed by generating  $N_s$  particles in the topological nodes that are in the radius of  $\sigma^2$  of initial GNSS reading  $g_0$ . Thus, generating  $\mathbf{P}_0$ , the set of particles at  $t = 0$ . The value of  $\sigma^2$  determines the spread of particles across the topological nodes in the first time-step. A more relaxed approach is to generate  $N_s$  particles across all topological nodes.

**1) Prediction:** Generate  $\hat{\mathbf{P}}_t$  from  $\mathbf{P}_t$  using  $\mathbf{R}_t$ .

*Explanation:* In the prediction step, all particles  $p_t^i(\cdot)$ , are updated to  $\hat{p}_t^i(\cdot)$  according to their probabilities of transition  $r_i(\tau_i^{(\cdot)}) \in \mathbf{R}_t$ . For example, a particle  $p_t^i(j)$  at node  $n_j$  is updated to  $\hat{p}_t^i(k)$  at node  $n_k$ , according to its probability of transition  $r_i(\tau_i^j)$  if  $n_j$  and  $n_k$  are connected through an edge in  $E$  i.e.,  $e_{jk} \neq 0$ . Note that if the transition probability is too low (small  $\tau_i^{(\cdot)}$ ), the predicted particles will remain at the same topological node. Hence,  $\hat{p}_t^i(\cdot) = p_t^i(\cdot)$ .

**2) Weighting:** Generate  $\mathbf{W}_t$  by calculating  $w_t^i \propto \|\hat{p}_t^i(\cdot) - g_t\| \forall \hat{p}_t^i(\cdot) \in \hat{\mathbf{P}}_t$ .

*Explanation:* Normalised weights are calculated for all predicted particles. These weights are proportional to the distance between the GNSS location and predicted samples.

**3) Update:** Calculate  $\mathcal{T}_{(\cdot)} = \sum_{i \in n_{(\cdot)}} (\hat{p}_t^i(\cdot) w_t^i) \forall n_{(\cdot)} \in N$ .

*Explanation:* The sum of weighted predicted particles occupying a topological node gives the node a topological mass  $\mathcal{T}$ . The coordinates of the topological node with the largest topological mass are considered as the estimated location by TPF:

$$\text{TPF output} = \max_{n_{(\cdot)}} (\mathcal{T}_{(\cdot)}). \quad (3)$$

**4) Resample and reiterate:** Sample from  $\hat{\mathbf{P}}_t$  with probability  $\mathbf{W}_t$  allowing repetition, thus generating  $\mathbf{P}_{t+1}$ . Go to Step 1.

*Explanation:* New samples are drawn with repetition, from the predicted samples, where the probability of selection is proportional to the weights calculated in Step 2. This new set of samples are fed back as an input to TPF in the prediction step.

## V. EXPERIMENTAL SETUP

Our framework has been fully integrated into the Robot Operating System (ROS) [20]. The data collected and experiment performed in this work were carried out on Ubuntu 16.04 LTS with ROS Kinetic.

**1) Polytunnels:** Experiments were performed in operational polytunnels at Maidstone, Kent, UK. Each polytunnel is 135 m long with five rows of raised beds of strawberry plants and four lanes for the fruit pickers to navigate. The topological map consists of 46 nodes per lane, thus each polytunnel is divided into 184 discrete points. These polytunnels were installed with three Kodak Pixpro 360 camera which provided the overhead images and timestamps used to find the ground truth of the pickers at different points in the polytunnel.

**2) Thorvald Robotic Platform:** is a multi-purpose, highly modular and versatile agri-robotic platform (see Fig. 1). For the presented experiments, the width of the robots was adjusted to fit the narrow lanes of the polytunnels and the configuration adjusted to carry at least four trays of strawberry produce. Two Thorvald robots were used in these experiments for the transportation of picked strawberries from polytunnels to the FHU.

**3) Smart Trolley:** A fruit carrying trolley that could hold up to four trays of produce, equipped with a Garmin Glo 2, a GNSS module, a cellphone/tablet for the CAR application and a power bank. The Garmin Glo 2 was programmed to bypass the cellphone's internal GNSS. Thus, the CAR application could obtain GNSS coordinates directly from GLO 2 via a Bluetooth connection.

**4) Scenario:** Each fruit picker is provided with a smart trolley that can carry up to four trays of strawberries. Once the trays on the trolley are (almost) full, a robot can be requested via the CAR application. The summoned robot will already be loaded with empty fruit trays by workers at the FHU. Once the robot arrives at the location of the picker, it is stocked with full strawberry trays while the trolley is loaded with empty trays provided by the robot. The fruit picker can now keep on picking while the robot transfers the full strawberry trays to FHU. For this cycle to work, it is imperative that the robot arrives at the correct lane in the polytunnel. Failure of which will require the fruit picker to carry full trays of produce to the robot. This will require the fruit pickers to exit the polytunnel and enter the lane where the robot is waiting to be loaded, as the pickers cannot jump to a neighbouring lane directly due to the obstruction provided by the raised beds carrying strawberry plants.

**5) Results:** These experiments were performed in real operational polytunnels utilising two Thorvald robots, two smart trolleys and two human pickers. The topological map for the polytunnels used during the experiments is shown in Fig. 4. The black edges connect the topological nodes of the same lane, while the white edges tie two neighbouring lanes together at the front and rear of the tunnel. The white edges are necessary to track the picker when they are making a transition from one lane to another.

Fig. 3a and 3b, show the pickers picking strawberries in lane D and lane C, respectively. The raw GNSS locations are also plotted as blue markers, while the green markers are the outputs of TPF. The red markers represent the closest nodes to the raw GNSS measurements. Feeding the raw GNSS measurements to the robots will result in the robot navigating

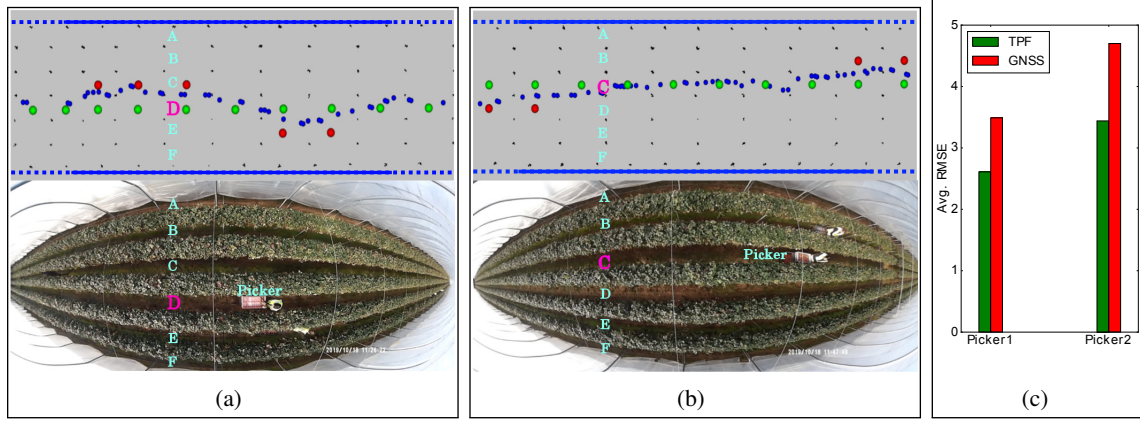


Fig. 3: Markers: TPF (green), GNSS closest node (red), raw GNSS (blue). Parameters:  $\lambda = 0.1$ ,  $N_s = 250$ . Fruit picker at lane D (a) and lane C (b). (c) Avg. RMSE of TPF and closest node using raw GNSS measurements inside a polytunnel.

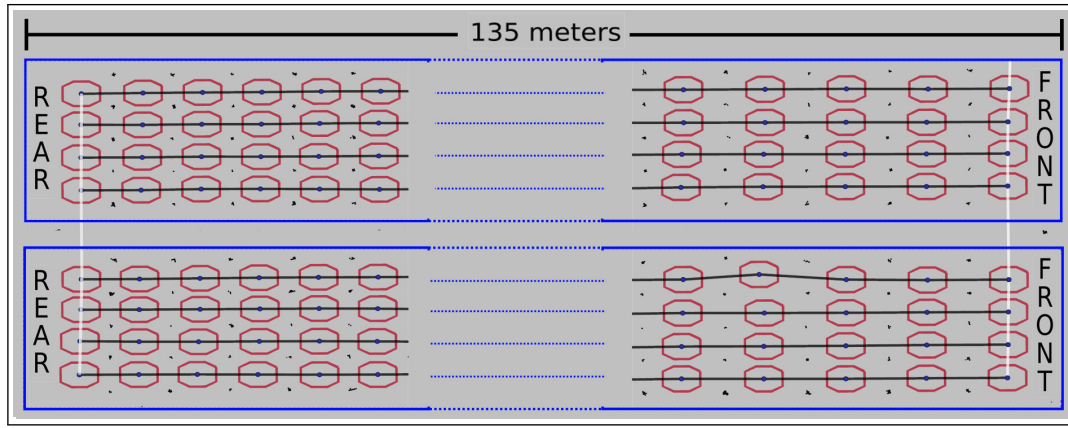


Fig. 4: Topological map of two polytunnels used in the experiments.

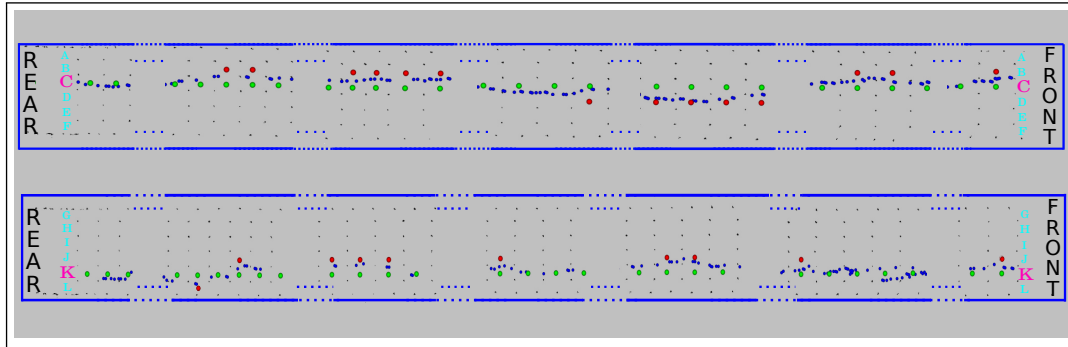


Fig. 5: Markers: Green-TPF, Red-GNSS based closest node, Blue-Raw GNSS. Parameters:  $\lambda = 0.1$ ,  $N_s=300$ . TPF performance comparison with raw GNSS for 135 meter long lanes of two polytunnels. Picking lanes: C and K.

to the wrong lane at certain timestamps. While the observed TPF output always stays in the correct picking lane.

In Fig. 3c, the average root mean square error (avg. RMSE) is obtained for TPF and compared with the raw GNSS-based closest node approach. The avg. RMSE is calculated by comparing the estimates of TPF with the ground truth location of the pickers obtained with the overhead cameras. Maps of two polytunnels with markers representing TPF, GNSS based closest node and raw GNSS measurements

are plotted in Fig. 5. The TPF correctly gets initialised in lane C and lane K of the polytunnels and stays in this lane despite the consistent erroneous measurements obtained via GNSS device. The closest node approach fails at multiple instances, providing the system with picker's position in lane B, D, J and L which were unoccupied.

A likely scenario is when the TPF gets initialised in a lane where the picker is not present. This will render all the samples of TPF to navigate between the topological nodes



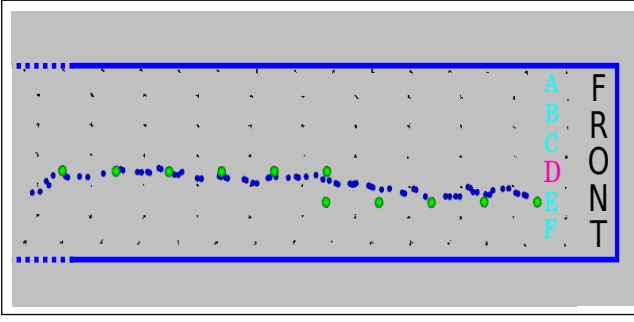


Fig. 6: TPF recovery from wrong initialisation. Parameters:  $\lambda = 0.1$ ,  $N_s=250$ . Markers: Green-TPF, Blue-Raw GNSS.

that belong to an unoccupied lane of the polytunnel. To solve this problem, the TPF only allow a small number particle to navigate freely without following the constraint set by the topological map. This results in the output of TPF recover to the correct lane of the polytunnel after a few updates. This behaviour is observed in Fig. 6, where the picker is picking in lane D but due to noisy first GNSS readings the TPF gets initialised in lane E. The recovery of TPF to lane D can be observed after a few iterations.

## VI. CONCLUSIONS

This paper proposes a novel discrete space continuous-time particle filter which exploits the topology of the environment to constrain the state transition of the particle filter. The proposed filter is deployed for tracking fruit pickers inside polytunnels. The resultant filter shows considerable improvement over the GNSS-based closest topological node approach. Promising results are obtained through experiments performed in fully operational strawberry polytunnels.

One of the observed issue with TPF is its failure to detect the correct lane in case of consistent bias in the GNSS data. This issue, however, can be resolved in a number of ways. One can dynamically estimate the bias in GNSS data by utilising a sensor like a lidar on the robot. As the robot approaches a picker the data from the lidar can be used to obtain an accurate estimate of pickers location for a brief period of time and for calculating the GNSS bias which can be removed prior to feeding the data to TPF. Another method to detect the correct picking lane is by tagging the front of every lane with an RFID chip. The chips can be then used to detect pickers when entering a lane of the polytunnel. Once the lane is detected, a hard constraint can be imposed on the samples of TPF that will cease the particles to jump to a topological node that does not belong to the detected lane, even in the presence of biased GNSS measurements. Furthermore, the GNSS data can be passed through a bias reducing algorithm as presented in [22], [23].

## ACKNOWLEDGEMENT

This work is part of the RAS-Berry (Robotic and Autonomous Systems for Berry Production, <https://rasberryproject.com/>) research programme, partly funded by Innovate UK under grant agreement No. 103339.

## REFERENCES

- [1] Duckett, T., Pearson, S., Blackmore, S. and Grieve, B. "Agricultural robotics: The future of robotic agriculture". Tech. Rep., The UK Robotics and Autonomous Systems Network (UK-RAS) (2018).
- [2] C. Cadena et al., "Past, present, and future of simultaneous localization and mapping: Toward the robust-perception age," *IEEE Trans. Robot.*, vol. 32, no. 6, pp. 1309–1332, Dec. 2016.
- [3] G. Bresson, Z. Alsayed, L. Yu and S. Glaser, "Simultaneous Localization and Mapping: A Survey of Current Trends in Autonomous Driving," in *IEEE Transactions on Intelligent Vehicles*, vol. 2, no. 3, pp. 194–220, Sept. 2017.
- [4] Wang, Y., Chen, Q., Zhu, Q., Liu, L., Li, C., Zheng, D. "A Survey of Mobile Laser Scanning Applications and Key Techniques over Urban Areas", in *MDPI Remote Sensing*, 11, 1540, 2019.
- [5] P. Dickinson, G. Cielniak, O. Szymanezyk and M. Mannion, "Indoor positioning of shoppers using a network of Bluetooth Low Energy beacons," *International Conference on Indoor Positioning and Indoor Navigation (IPIN)*, Alcalá de Henares, pp. 1–8, 2016.
- [6] P. From, L. Grimstad, M. Hanheide, S. Pearson, and G. Cielniak, "RASberry - Robotic and Autonomous Systems for Berry Production," *Mechanical Engineering Magazine Select Articles*, vol. 140, no. 6, pp. 14–18, 2018.
- [7] D. F. Glas, T. Kanda, H. Ishiguro and N. Hagita, "Simultaneous people tracking and localization for social robots using external laser range finders", *RSJ Int. Conf. on Intelligent Robots and Systems*, St. Louis, pp. 846–853, 2009.
- [8] Glas, D.F., Morales, Y., Kanda, T. et al. "Simultaneous people tracking and robot localization in dynamic social spaces", *Autonomous Robots* vol. 39, pp: 43–63, 2015.
- [9] Kim, M., et al. "Face tracking and recognition with visual constraints in real-world videos." In *Proc. CVPR 2008*, pp: 1–8.
- [10] Lu, W., et al. "A color histogram based people tracking system." In *Proc. ISCAS 2001*, pp: 137–140.
- [11] Park, U., et al. "Face tracking and recognition at a distance: A coaxial and concentric PTZ camera system", *IEEE Trans. on Information Forensics and Security*, 2013, 8(10), pp: 1665–1677.
- [12] Bellotto, N., et al. "Multisensor data fusion for joint people tracking and identification with a service robot." In *Proc. ROBIO 2007*.
- [13] Bellotto, N., Cosar, S., Yan, Z.: Human detection and tracking. In: Ang, M.H., Khatib, O., Siciliano, B. (eds.) *Encyclopedia of Robotics*, pp. 1–10. Springer (2018)
- [14] A. P. Gritti, O. Tarabini, J. Guzzi, G. A. D. Caro, V. Caglioti, L. M. Gambardella, and A. Giusti, "Kinect-based people detection and tracking from small-footprint ground robots," in *International Conference on Intelligent Robots and Systems (IROS)*, 2014.
- [15] Li X., Zhang Y., Marsic I., Burd R. S. Online People Tracking and Identification with RFID and Kinect, 2017. [arXiv: https://arxiv.org/pdf/1704.04861.pdf](https://arxiv.org/pdf/1704.04861.pdf)
- [16] Z. Yan, T. Duckett, and N. Bellotto, "Online learning for human classification in 3d LIDAR-based tracking," in *Proc. IEEE/RSJ Int. Conf. Intell. Robots Syst.*, 2017, pp. 864–871.
- [17] M. Imperoli, C. Potena, D. Nardi, G. Grisetti and A. Pretto, "An Effective Multi-Cue Positioning System for Agricultural Robotics," in *IEEE Robotics and Automation Letters*, vol. 3, no. 4, 2018.
- [18] Bellotto, N., Hu, H. "Computationally efficient solutions for tracking people with a mobile robot: an experimental evaluation of Bayesian filters", *Autonomous Robots* vol. 28, pp: 425–438, 2010.
- [19] M. W. Khan, N. Salman, A. Ali, A. M. Khan and A. H. Kemp, "A comparative study of target tracking with Kalman filter, extended Kalman filter and particle filter using received signal strength measurements," *Int. Conf. Emerg. Tech. (ICET)*, Peshawar, pp: 1–6, 2015.
- [20] M. Quigley, K. Conley, B. P. Gerkey, J. Faust, T. Foote, J. Leibs, R. Wheeler, and A. Y. Ng, "ROS: an open-source robot operating system," in *ICRA Workshop on Open Source Software*, 2009.
- [21] A. Binch, G.P. Das, J.P. Fentanes, and M. Hanheide, "Context Dependent Iterative Parameter Optimisation for Robust Robot Navigation," (Accepted to be) in *Proc. 2020 IEEE Int. Conf. Robots Autom. (ICRA)*, 31 May–4 Jun. 2020.
- [22] B. Townsend and P. Fenton, "A practical approach to the reduction of pseudorange multipath errors in a L1 GPS receiver," in *Proc. Int. Tech. Meeting Satell. Division Inst. Navig. ION GPS*, Sep. 20–23, 1994.
- [23] Frair, J.L., Nielsen, S.E., Merrill, E.H., Lele, S.R., Boyce, M.S., Munro, R.H., Stenhouse, G.B. and Beyer, H.L., "Removing GPS collar bias in habitat selection studies", *Journal of Applied Ecology*, vol. 41 no. 2, pp: 201–212, 2004.